Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method, comprising:

in an object oriented run-time environment, after a classfile has been loaded:

a) invoking a second method from a first method, said first method belonging to said class, said invoking comprising providing an identification of said first method and said classfile;

b) identifying a plug-in module for said first method based upon said identification, said plug-in module containing a handler method;

c) executing said handler method to report and/or record information about said first method;

d) executing said first method from a point beyond where said second method was invoked;

e) flowing from said first method to a third method;

f) invoking said second method from said third method, said invoking including providing an identification of said third method and a second classfile that said third method is a part of, said second classfile having been loaded at least by the completion of e) above ;

g) identifying said plug-in module for said third method based upon said third method and second classfile identification;

h) executing said handler method to report and/or record information about said third method; and,

i) executing a portion of said third method from a point beyond where said second method was invoked from said third method, wherein said classfile and said second classfile were both modified, prior to their respectively being said loaded, with additional bytecode instructions that perform a) and f) above.

2. (Previously Presented) The method of claim 1 wherein said executing of said handler method at c) above causes an entry time for said first method to be recorded.

3. (Previously Presented) The method of claim 1 wherein said executing of said handler method at c) above causes an exit time for said first method to be recorded.

4. (Previously Presented) The method of claim 1 wherein said executing of said handler method at c) above causes a counter maintained for said first method to be incremented.

5. (Previously Presented) The method of claim 1 wherein said executing of said handler method at c) above causes an input parameter value of said first method to be recorded.

6. (Previously Presented) The method of claim 1 wherein said executing of said handler method at c) above causes a returned value of said first method to be recorded.

7. (Original) The method of claim 1 wherein said first method is a constructor.

8. (Previously Presented) The method of claim 1 further comprising creating, prior to said invoking at a) above, an object having an input parameter value of said first method.

9. (Previously Presented) The method of claim 1 wherein said invoking at a) above further comprises providing an input parameter value of said first method.

10. (Previously Presented) The method of claim 1 wherein said invoking at a) above further comprises identifying where said first method's instructions can be found in memory.

11. (Previously Presented) The method of claim 1 further comprising, after said executing said first method from a point beyond where said second method was invoked but before said flowing to said third method at e) above:

invoking a third method from said first method because said first method is about to reach an exit point, said second method having been invoked from said first method because an entry point of said first method had just been reached;

re-identifying said plug-in module for said first method as a consequence of said invoking a third method;

re-executing said handler method to report and/or record information about said first method; and,

executing a remaining portion of said first method through said exit point.

12. (canceled)

13. (Previously Presented) The method of claim 1 wherein g) further comprises also identifying a second plug-in module for said third method based upon said third method and second class identification, said second plug-in module containing a second handler method.

14. (Previously Presented) The method of class 13 further comprising also executing said second handler method to report and/or record different information about said third method than what said first handler method reports ~~r~~ and/or records about said third method.

15. (Previously Presented) The method of claim 14 wherein a first object is called to execute said first method and a second object is called to execute said third method.

16. (Previously Presented) The method of claim 15 wherein said object oriented run-time environment is a Java object oriented environment.

17. (Previously Presented) The method of claim 1 wherein said invoking at a) above further comprises providing said first method's signature, said first method's signature comprising:

said identification of said first method;

said identification of said class that said first method is a part of; and,

said first method's arguments.

18. (Currently Amended) An article of manufacture having stored thereon executable or interpretable program code which when processed by one or more computing systems cause a method to be performed, said method, comprising:

in an object oriented run-time environment, after a classfile has been loaded:

a) invoking a second method from a first method, said first method belonging to said class, said invoking comprising providing an identification of said first method and said classfile;

b) identifying a plug-in module for said first method based upon said identification, said plug-in module containing a handler method;

c) executing said handler method to report and/or record information about said first method;

d) executing said first method from a point beyond where said second method was invoked;

e) flowing from said first method to a third method;

f) invoking said second method from said third method, said invoking including providing an identification of said third method and a second

classfile that said third method is a part of, said second classfile having been loaded at least by the completion of e) above;

g) identifying said plug-in module for said third method based upon said third method and second classfile identification;

h) executing said handler method to report and/or record information about said third method; and,

i) executing a portion of said third method from a point beyond where said second method was invoked from said third method, wherein said classfile and said second classfile were both modified, prior to their respectively being said loaded, with additional bytecode instructions that perform a) and f) above.

19. (Previously Presented) The article of manufacture of claim 18 wherein said executing of said handler method at c) above causes an entry time for said first method to be recorded.

20. (Previously Presented) The article of manufacture of claim 18 wherein said executing of said handler method at c) above causes an exit time for said first method to be recorded.

21. (Previously Presented) The article of manufacture of claim 18 wherein said executing of said handler method at c) above causes a counter maintained for said first method to be incremented.

22. (Previously Presented) The article of manufacture of claim 18 wherein said executing of said handler method at c) above causes an input parameter value of said first method to be recorded.

23. (Previously Presented) The article of manufacture of claim 18 wherein said executing of said handler method at c) above causes a returned value of said first method to be recorded.

24. (Previously Presented) The article of manufacture of claim 18 wherein said first method is a constructor.

25. (Previously Presented) The article of manufacture of claim 18 further comprising creating, prior to said invoking at a) above, an object having an input parameter value of said first method..

26. (Previously Presented) The article of manufacture of claim 18 wherein said invoking at a) above further comprises providing an input parameter value of said first method.

27. (Previously Presented) The article of manufacture of claim 18 wherein said invoking at a) above further comprises identifying where said first method's instructions can be found in memory.

28. (Previously Presented) The article of manufacture of claim 18 further comprising, after said executing said first method from a point beyond where said second method was invoked but before said flowing to said third method at e) above:

invoking a third method from said first method because said first method is about to reach an exit point, said second method having been invoked from said first method because an entry point of said first method had just been reached;

re-identifying said plug-in module for said first method as a consequence of said invoking a third method;

re-executing said handler method to report and/or record information about said first method; and,

executing a remaining portion of said first method through said exit point.

29. (Canceled).

30. (Previously Presented) The article of manufacture of claim18 wherein g) further comprises also identifying a second plug-in module for said third method based upon said third method and second class identification, said second plug-in module containing a second handler method.

31. (Previously Presented) The article of manufacture of claim 30 further comprising also executing said second handler method to report and/or record

different information about said third method than what said first handler method reports and/or records about said third method.

32. (Previously Presented) The article of manufacture of claim 31 wherein a first object is called to execute said first method and a second object is called to execute said third method.

33. (Previously Presented) The article of manufacture of claim 32 wherein said object oriented run-time environment is a Java object oriented environment.

34. (Previously Presented) The article of manufacture of claim 18 wherein said invoking at a) above further comprises providing said first method's signature, said first method's signature comprising:

said identification of said first method;

said identification of said class that said first method is a part of; and,

said first method's arguments.

35. (Previously Presented) A computing system, comprising:

a first classfile;

a dispatcher, said dispatcher having a dictionary

a first object manufactured from said classfile, said first object having a first method, said first method instrumented with first program code to invoke a second method executed by said dispatcher, said first program code written to identify said

classfile and said first method to said dispatcher as part of said invoking, said first program code located proximate to said first method's entry point;

a plug-in, said plug-in having a handler, said handler having program code to report and/or record information about a method that invokes said dispatcher, said dispatcher's dictionary correlating said first method and said classfile with said plug-in;

a second classfile;

a second object manufactured from said second classfile, said second object having a third method, said third method instrumented with second program code to invoke said second method executed by said dispatcher, said second program code written to identify said second classfile and said third method to said dispatcher as part of said third method's invoking of said dispatcher, said second program code located proximate to said third method's entry point, said dispatcher's dictionary correlating said third method and said second classfile with said plug-in.

36. (Previously Presented) The computing system of claim 35 wherein said invocation of said second method by said first program code also includes identifying said first method's arguments.

37. (Previously Presented) The computing system of claim 36 wherein said invocation of said second method by said second program code also includes identifying said third method's arguments.

38. (Previously Presented) The computing system of claim 37 wherein said first and second objects are Java objects.

39. (Previously Presented) The computing system of claim 35 wherein said first and second objects are Java objects.

40. (Previously Presented) The computing system of claim 35 wherein said first method is a constructor.

41. (Previously Presented The computing system of claim 35 wherein said information includes a time of entry or exit of said method.

42. (Previously Presented) The computing system of claim 35 wherein said information includes incrementing a counter maintained for said method.

43. (Previously Presented) The computing system of claim 35 wherein said information includes a value for an argument of said method.